Project Plan: Instance FeedbackApp

The goal for this project is to develop a feedback app where teachers can gather feedback from their lectures. The primary objectives are to develop the project according to the project owner's requirements and to implement an application that will enable teachers to improve their lectures with the help of the feedback they collect.

Scope Definition

As this project is carried out as a part of a course, there are not enough resources for a big scope of features and functions. The lack of resources has been taken into consideration when defining the scope for this project. The core features and functionalities that are implemented for the first version of the application include the following:

- User authentication.
- Different roles depending on if the user is a teacher or a student.
- Enrolling in and leaving a course and giving feedback to a lecture for students.
- Creating courses and lectures and viewing the feedback given to the lectures for users that are teachers.
- Removing students from courses.
- Editing user profile for all users.

The project could be scaled to have more features and functions that were left out of the first version. The features that were left out are as follows:

- Data visualization: graphical formatting of data using, for example, different kinds of charts.
- Statistics of feedback from any given period, for example, from one month or week. The excluded features will be implemented in the second version of the application.

Requirements Gathering

As this project is conducted as a part of a course, and the product owner is the teacher, there is no formal process for gathering requirements. The product owner has given this project requirements, and there have been discussions regarding the backlog for the project. For the functional requirements, the primary function is to develop an application that will enable teachers to get feedback from the students. The other functional requirements are discussed below:

- Teachers can create and manage courses and create and manage lectures within the courses. They can also approve or reject requests to join the courses and review feedback given by the students.
- Students, on the other hand, can join courses and give feedback for the lectures they attend.

Resource Allocation

The necessary resources include four students as personnel, computers for each student, and software for developing the application. The students use their own computers. The software for developing includes a free software visual studio code or webstorm, which is commercial software. The license for webstorm is provided to all students by Metropolia University of Applied Sciences. There is no budget for the project.

Risk Assessment

The risks for the project are minimal as there is no budget included. In case the project fails, the risk is the students not passing the course, having to retake it later.

Team Formation

The team for the project was formed from students attending the Ohjelmistoprojekti course. The students could form the teams by themselves, which means that the team members agreed to work together for the project without input from the teacher.

Technology Stack Selections

Frontend Technologies

- React: Used for building user interfaces.
- JavaScript: Core programming language for front-end and backend.
- Vitest: Framework for testing React components.
- Tailwind CSS: Utility-first CSS framework for custom designs.
- Redux & Redux Toolkit: State management tools.
- Axios: HTTP client for requests to backend.

Backend Technologies

- Node.js: JavaScript runtime for server-side.
- Express.js: Web framework for routing and middleware.
- Prisma ORM: Database toolkit for interacting with PostgreSQL.
- PostgreSQL: Object-relational database system.
- Jest: Testing framework

Sequencing of Activities

1. Project Initialization: Setup version control, repositories, and project directories.
2. Environment Setup: Configure development, testing, and production environments, setup database and ORM.
3. Backend Development: Define and implement API endpoints, authentication, and authorization.
4. Frontend Development: Setup React, Redux, Axios, and implement user interfaces using Tailwind CSS.
5. Testing: Write and conduct unit, integration, and manual tests using Vitest and Jest.
6. Deployment: Configure servers, deploy application, setup CI/CD pipelines.

Communication Plan

Communication for the project happens at school when the team members are present, and other times via Discord. Project meetings are held every Monday and Wednesday. The meetings are held either at school or in Discord. The team has a Discord server set up, and the members can discuss the project there.

Budgeting and Cost Estimation

As mentioned earlier, this project has a budget of 0 euros as it is a school project.

Client Involvement

Client involvement for this project is maintained by communicating with the project owner during sprint meetings every Monday. In terms of legal and compliance considerations, the team adheres to relevant school or institutional policies and standards that apply to the project.